Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A computer-implemented method for identifying user interface (UI) objects in a markup-language stream, the method comprising:

receiving, from a server and at a client computer system, a web-based application for display in a web browser, the web-based application comprising one or more web pages;

analyzing the web-based application at the client computer system;

defining an application-specific grammar at the client computer system, based on analysis of the web-based application;

automatically generating a parser computer program at the client computer system, based on the application-specific grammar, using an automated parser generator tool;

scanning document object model (DOM) of the web-based application with the parser computer program to generate tokens;

parsing the tokens with the parser computer program to identify at least one graphical element in the web-based application;

outputting, from the parser computer program at the client computer system to a context-based help utility at the client computer system, information about position and content of the at least one graphical element identified by parsing the tokens in the web-based application; and

providing context-based help based at least in part on the at least one graphical element in the web-based application.

2. (Original) The method of claim 1, wherein said markup-language stream drives a markup-language-based browser application, and wherein the scanning step includes scanning the DOM generated by a browser that displays that application.

3. (Original) The method of claim 1, wherein the scanning step includes identifying elements of the DOM by traversal thereof.

4. (Canceled)

5. (Previously Presented) The method of claim 3, wherein the scanning step includes generating one or more tokens for each scanned DOM element.

6.-7. (Canceled) .

8. (Currently Amended) The method of claim 1, wherein the at least one graphical element ~~UI objects~~ comprises one of a user input field, a text field, a metatag, unprintable markup-language, or an in-line image.

9. (Previously Presented) The method of claim 1, wherein the scanning and parsing steps are adapted to identify UI objects that correspond to elements displayed in the web-based application.

10. (Previously Presented) The method of claim 1, further comprising grouping the tokens into syntactic structures that identify items displayed by the web-based application.

11. (Previously Presented) The method of claim 10, wherein said step of grouping comprises identifying similarly formatted markup-language elements based on their markup-language attributes such as class name, font size, style, tag color, and size.

12. (Currently Amended) The method of claim 1, wherein said at least one graphical element ~~UI objects~~ comprises a name, content, a shape, or a location.

13. (Previously Presented) The method of claim 1, wherein automatically generating said the parser computer program comprises executing YACC ("Yet Another Compiler-Compiler").

14.-15. (Canceled).

16. (Previously Presented) The method of claim 1, wherein the parser computer program is a LALR(1) parser.

17. (Previously Presented) The method of claim 1, wherein the parser computer program is a LR(1) parser.

18. (Previously Presented) The method of claims 1, wherein the markup language is any of HTML, XHTML and XUL.

19. (Previously Presented) A digital data processing system comprising:

a client digital data processor at a client computer system, the client digital data processor being configured to:

receive, from a server and at the client computer system, a web-based application for display in a web browser, the web-based application comprising one or more web pages;

analyze the web-based application at the client computer system;

define an application-specific grammar at the client computer system, based on analysis of the web-based application;

automatically generate a parser computer program at the client computer system, based on the application-specific grammar, using an automated parser generator tool;

scan document object model (DOM) of the web-based application with the parser computer program to generate tokens;

parse the tokens with the parser computer program to identify at least one graphical element in the web-based application;

output, from the parser computer program at the client computer system to a context-based help utility at the client computer system, information about position and

content of the at least one graphical element identified by parsing the tokens in the web-based application; and

provide context-based help based at least in part on the at least one graphical element in the web-based application.

20. (Canceled)

21. (Currently Amended) The digital data processing system of ~~claim 20~~ claim 19, wherein said at least one graphical element ~~one or more UI objects~~ each comprise name, content, shape, location, and properties.

22. (Canceled).

23. (Previously Presented) The digital data processing system of claim 19, wherein said tokens are interpreted according to the application-specific grammar to identify and distinguish among UI objects of the web-based application's display.

24. (Currently Amended) The digital data processing system of claim 19, wherein the at least one graphical element ~~UI object~~ comprises a user input field, a text field, a metatag, unprintable markup-language, or an in-line image.

25. (Previously Presented) The digital data processing system of claim 19, wherein the markup language is any of HTML, XHTML and XUL.

26.-29. (Canceled)